

CLAIMS

1 1. A method for managing a data caching service from a management terminal in a
2 distributed computer system having a host computer system with a host memory
3 and at least one storage device connected to the computer system by driver
4 software, the method comprising:

- 5 (a) inserting an interface layer between the driver software and the storage
6 device, the interface layer exporting a platform dependent API and
7 controlling data passing between the driver software and the storage
8 device;
9 (b) running, in the host, management facade software that converts the
10 interface layer API to platform-independent method calls;
11 (c) running, in the host, a federated bean that generates method calls to the
12 management facade to control the interface layer; and
13 (d) controlling the federated bean to enable a data read cache by instructing
14 the interface layer to intercept requests for data from the storage device
15 and, if the data is in the host memory, to retrieve the data from the host
16 memory, and to retrieve the data from the storage device if the data is not
17 in the host memory.

1 2. The method of claim 1 wherein step (d) comprises controlling the federated bean
2 with a command line interface.

1 3. The method of claim 1 wherein step (d) comprises controlling the federated bean
2 with a graphical user interface.

1 4. The method of claim 1 further comprising:

- 2 (e) using the federated bean to disable the read cache.

- 1 5. The method of claim 1 further comprising:
2 (f) using the federated bean to set characteristics of the read cache.
- 1 6. The method of claim 1 wherein the computer system further comprises non-
2 volatile memory and the method further comprises:
3 (g) controlling the federated bean to enable a data write cache by instructing
4 the interface layer to intercept requests to write data to the storage device
5 and to write the data to the non-volatile memory.
- 1 7. The method of claim 6 further comprising:
2 (h) enabling a flusher thread to periodically copy data in the non-volatile
3 memory to the storage device.
- 1 8. The method of claim 7 wherein step (h) comprises enabling a plurality of flusher
2 threads wherein the number of enabled flusher threads is determined by the
3 federated bean.
- 1 9. The method of claim 6 further comprising:
2 (i) using the federated bean to disable the write cache.
- 1 10. The method of claim 6 further comprising:
2 (j) using the federated bean to set characteristics of the write cache.
- 1 11. Apparatus for managing a data caching service from a management terminal in a
2 distributed computer system having a host computer system with a host memory
3 and at least one storage device connected to the computer system by driver
4 software, the apparatus comprising:

an interface layer inserted between the driver software and the storage device, the interface layer exporting a platform dependent API and controlling data passing between the driver software and the storage device;

management facade software that runs in the host and converts the interface layer API to platform-independent method calls;

a federated bean that runs in the host and generates method calls to the management facade to control the interface layer; and

a presentation program that controls the federated bean to enable a data read cache by instructing the interface layer to intercept requests for data from the storage device and, if the data is in the host memory, to retrieve the data from the host memory, and to retrieve the data from the storage device if the data is not in the host memory.

12. The apparatus of claim 11 wherein the presentation program comprises a command line interface.

13. The apparatus of claim 11 wherein the presentation program comprises a graphical user interface.

14. The apparatus of claim 11 wherein the presentation program controls the federated bean to disable the read cache.

15. The apparatus of claim 11 wherein the presentation program controls the federated bean to set characteristics of the read cache.

16. The apparatus of claim 11 wherein the computer system further comprises non-volatile memory and wherein the federated bean comprises methods that enable a data write cache by instructing the interface layer to intercept requests to write data to the storage device and to write the data to the non-volatile memory.

- 1 17. The apparatus of claim 16 further comprising a flusher thread that periodically
2 copies data in the non-volatile memory to the storage device.
- 1 18. The apparatus of claim 17 further comprising a plurality of flusher threads
2 wherein the number of enabled flusher threads is determined by the federated
3 bean.
- 1 19. The apparatus of claim 16 wherein the federated bean comprises methods that
2 disable the write cache.
- 1 20. The apparatus of claim 16 wherein the federated bean comprises methods that
2 set characteristics of the write cache.
- 1 21. A computer program product for managing a data caching service from a
2 management terminal in a distributed computer system having a host computer
3 system with a host memory and at least one storage device connected to the
4 computer system by driver software, the computer program product comprising a
5 computer usable medium having computer readable program code thereon,
6 including:
7 an interface layer inserted between the driver software and the storage
8 device, the interface layer exporting a platform dependent API and controlling
9 data passing between the driver software and the storage device;
10 management facade software that runs in the host and converts the
11 interface layer API to platform-independent method calls;
12 federated bean software that runs in the host and generates method calls
13 to the management facade to control the interface layer; and
14 a presentation program that controls the federated bean to enable a data
15 read cache by instructing the interface layer to intercept requests for data from

16 the storage device and, if the data is in the host memory, to retrieve the data
17 from the host memory, and to retrieve the data from the storage device if the data
18 is not in the host memory.

1 22. The computer program product of claim 21 wherein the computer system further
2 comprises non-volatile memory and wherein the federated bean comprises
3 methods that enable a data write cache by instructing the interface layer to
4 intercept requests to write data to the storage device and to write the data to the
5 non-volatile memory.

1 23. The computer program product of claim 22 further comprising program code that
2 creates a flusher thread that periodically copies data in the non-volatile memory
3 to the storage device.

1 24. The computer program product of claim 23 further comprising program code for
2 creating a plurality of flusher threads wherein the number of enabled flusher
3 threads is determined by the federated bean.

1 25. A computer data signal embodied in a carrier wave for managing a data caching
2 service from a management terminal in a distributed computer system having a
3 host computer system with a host memory and at least one storage device
4 connected to the computer system by driver software, the computer program
5 data signal comprising:

6 program code for creating an interface layer inserted between the driver
7 software and the storage device, the interface layer exporting a platform
8 dependent API and controlling data passing between the driver software and the
9 storage device;

10 management facade software that runs in the host and converts the
11 interface layer API to platform-independent method calls;

12 federated bean software that runs in the host and generates method calls
13 to the management facade to control the interface layer; and
14 a presentation program that controls the federated bean to enable a data
15 read cache by instructing the interface layer to intercept requests for data from
16 the storage device and, if the data is in the host memory, to retrieve the data
17 from the host memory, and to retrieve the data from the storage device if the data
18 is not in the host memory.

09975465.101101
TOTOT"59457660